

(19)



Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 0 895 164 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
03.02.1999 Bulletin 1999/05

(51) Int. Cl.⁶: G06F 12/14

(21) Application number: 98114146.8

(22) Date of filing: 29.07.1998

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 01.08.1997 US 904991

(71) Applicant: MOTOROLA, INC.
Schaumburg, IL 60196 (US)

(72) Inventors:
• Johns-Vano, Kerry Lucille
Scottsdale, Arizona 85254 (US)
• Harrison, David Michael
Mesa, Arizona 85203 (US)

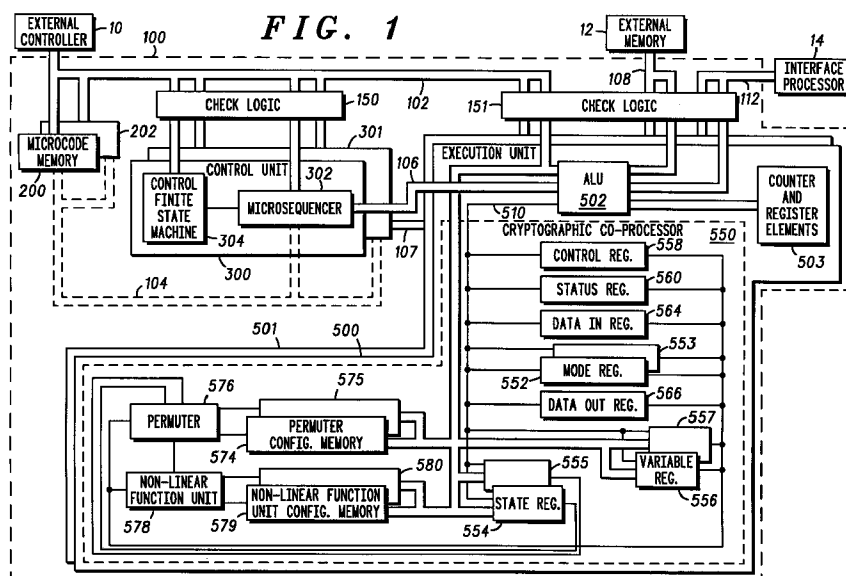
• Carswell, Phillip Anthony
Garland, Texas 75034 (US)
• Perea, William Louis
Chandler, Arizona 85226 (US)
• Lien, Ty Bao
Tempe, Arizona 85283 (US)

(74) Representative:
Potts, Susan Patricia et al
Motorola European Intellectual Property
Operations
Midpoint
Alencon Link
Basingstoke, Hampshire RG21 7PL (GB)

(54) Configurable cryptographic processing engine and method

(57) A configurable cryptographic engine (100) provides high performance cryptographic processing support for symmetric combiner type cryptographic algorithms. As many as two independent cryptographic algorithms may be performed at the same time through the processes of background staging and algorithm

multi-tasking. A 3-stage instruction pipeline, dynamically configurable cryptographic co-processor (550), and 32-bit RISC based architecture support high performance cryptographic processing performance on the order of 60 Mbps aggregate throughput.



EP 0 895 164 A2

Description

Field of the Invention

[0001] This invention relates in general to the field of secure cryptographic communications.

Background of the Invention

[0002] Trends in the communications market have clearly defined the need for security for both commercial and military markets. As communications systems become more sophisticated with complex communication services and capabilities, it is important to keep information secure. One of the problems with secure equipment is interoperability between existing and future communications products. As new products are introduced and expected to become part of a larger secure communications system, the need for integrating legacy and emerging technology becomes imperative. Another problem with modern communications equipment is that secure hardware implementations are commonly difficult to design and expensive to manufacture. The manufacture of secure hardware often requires special semiconductor foundries which operate in an expensive, high security environment. Another problem with hardware implementations of secure communications equipment is the difficulty in reconfiguring the hardware. Typical hardware implementations are difficult to reconfigure and reprogram when equipment must perform various functional activities in a non-homogeneous communications environment.

[0003] Communications equipment implemented with software, however, are typically not considered as secure as hardware implementations because of the accessibility of the software. A typical problem with software implementations is that concurrent processing of multiple programs results in performance loss due to program swapping in a secure operating system.

[0004] Thus, what is needed are an improved cryptographic processing engine and method suitable for use in cryptographic systems. What is also needed are a cryptographic processing engine and method that provides interoperability between existing and emerging communications technologies. What is also needed is a cryptographic processing engine that does not include cryptographic programs during manufacture and can be fabricated in commercial semiconductor foundries thereby reducing costs. What is also needed is a cryptographic engine and method which are configurable and programmable and capable of performing various and changeable communications functions. What is further needed is a cryptographic engine and method which provide a high security software component with limited accessibility. What is also needed is a cryptographic engine and method which rapidly and securely switches programs and context for each data member processed.

Brief Description of the Drawings

[0005] The invention is pointed out with particularity in the appended claims. However, a more complete understanding of the present invention may be derived by referring to the detailed description and claims when considered in connection with the figures, wherein like reference numbers refer to similar items throughout the figures, and:

FIG. 1 illustrates a hardware block diagram of a configurable cryptographic engine (CCE) in accordance with a preferred embodiment of the present invention;

FIG. 2 is a flow chart of a CCE setup and configuration procedure in accordance with a preferred embodiment of the present invention;

FIG. 3 is a flow chart of a data unit processing procedure in accordance with a preferred embodiment of the present invention.

[0006] The exemplification set out herein illustrates a preferred embodiment of the invention in one form thereof, and such exemplification is not intended to be construed as limiting in any manner.

Detailed Description of the Drawings

[0007] The present invention provides, among other things, a configurable processing engine and method. The present invention also provides a cryptographic processing engine and method suitable for providing interoperability between existing and emerging communications technology. The present invention also provides a cryptographic processing engine suitable for manufacture in commercial semiconductor foundries thereby reducing manufacturing costs. The present invention also provides a cryptographic engine and method which is configurable, programmable, and capable of performing various and changeable communications functions. The present invention also provides a cryptographic engine and method which provide a high security software component with limited accessibility. The present invention further provides a cryptographic engine and method which rapidly and securely switches programs and context for each data member processed.

[0008] FIG. 1 illustrates a hardware block diagram of a configurable cryptographic engine (CCE) in accordance with a preferred embodiment of the present invention. In the preferred embodiment, CCE 100 includes microcode memories 200 & 202, control units 300 & 301, execution units 500 & 501, and check logic 150 & 151.

[0009] Microcode memories 200 & 202 are coupled to external controller 10, check logic 150 & 151, control units 300 & 301, and execution units 500 & 501 through interface and control bus 102. Microcode memories 200

& 202 are also coupled to control units 300 & 301 through address and data bus 104. Each microcode memory may perform either as an "active" and "shadow" microcode memory element.

[0010] Active and shadow are titles associated with selected elements of CCE 100 which are related to procedures associated therewith. For example, when CCE 100 is being setup and configured to perform an operation, this procedure is performed in "background staging" mode. Background staging is a mode where a setup and configuration procedure is performed which prepares CCE 100 to process data. Elements, such as memory, registers, and others, which are associated with background staging mode are titled "shadow" elements. Elements which are titled shadow elements are determined preferably by external controller 10. Alternatively, an example of "active" mode elements is when shadow elements previously associated with the background staging mode are currently being used in processing data. More specifically, elements titled shadow during a setup and configuration procedure are titled active when those same elements are used in processing data. Other examples of active and shadow elements are when microcode memory 200 is titled the shadow memory and microcode memory 202 is titled the active memory and vice versa.

[0011] Control units 300 & 301 are coupled to external controller 10, microcode memories 200 & 202, check logic 150 & 151, and execution units 500 & 501 through interface and control bus 102. Control units 300 & 301 are also coupled to execution units 500 & 501 through status and control bus 106 & 107, respectively. In the preferred embodiment, control units 300 & 301 each contain microsequencer 302. In another embodiment, control units 300 & 301 also each contain control finite state machine 304 (CFSM). In an embodiment having CFSM 304, CFSM 304 is comprised of hardware control logic and software which, among other things, monitors for alarms and provides high level control of microsequencer 302. In the preferred embodiment, control unit 300 and redundant control unit 301 each contain microsequencer 302 which provides software control of CCE 100. Software control may also provide, for example, monitors for alarms, status and other types of operations. Preferably, control units 300 & 301 each perform redundant functionality of the other.

[0012] Execution units 500 & 501 are coupled to external controller 10, microcode memories 200 & 202, control units 300 & 301, and check logic 150 & 151 through interface and control bus 102. Execution units 500 & 501 are also coupled to control units 300 & 301 through 106 & 107, respectively. Execution units 500 & 501 are also coupled to external memory 12 through address and RAM bus 108. Furthermore, execution units 500 & 501 are coupled to interface processor 14 through interface processor bus 112. In the preferred embodiment, execution units 500 & 501 each contain Arithmetic Logic Unit 502 (ALU), counter and register elements 503, and

cryptographic co-processor 550 (CCO). In the preferred embodiment, ALU 502, counter and register elements 503, and CCO 550 are internally coupled through output bus 510. Preferably, execution units 500 & 501 each perform redundant functionality of the other. Additionally, the combination of microsequencer 302, ALU 502, and counter and register elements 503 function like a 32-bit RISC processor.

[0013] CCE 100 is a dynamically configurable cryptographic processing engine which, when configured, performs cryptographic operations on data units, e.g., preformatted data packets of information. CCE 100 is setup and configured by external controller 10. Preferably, external controller 10 performs a setup and configuration procedure for a channel program which prepares CCE 100 for processing data units. A channel program is characterized by a set of software program instructions which perform, among other things, cryptographic operations on a data unit. A channel program is preferably created by means external to the CCE. External controller 10 preferably provides external control when preparing CCE 100 for processing data units. A data unit is preferably stored in external memory 12 prior to processing. Following the processing of data units, a data unit is preferably written to external interface processor 14.

[0014] In the preferred embodiment, external controller 10 performs a setup and configuration procedure for a channel program. While performing this setup and configuration procedure, external controller 10 loads a channel program from external memory 12 into the shadow microcode memory, either memory 200 or 202. External controller 10 determines the microcode memory to receive a channel program. In the preferred embodiment, external controller 10 determines which microcode memory 200 or 202 is the shadow microcode memory to receive a channel program.

[0015] External controller 10 provides control over control units 300 & 301. Preferably, control unit 300 and redundant control unit 301 execute the same channel program from the active microcode memory. Control units 300 & 301 are comprised of functionally similar hardware and software although each control unit may contain hardware and software which is the logical complement of the other. A first check logic 150 examines outputs from control units 300 & 301 to provide an inspection of output values. In the preferred embodiment, first check logic 150 provides an alarm to external controller 10 when control units 300 & 301 produce conflicting output values. CCE 100 will also function without the redundant control unit and redundant execution unit.

[0016] Among other things, control units 300 & 301 each contain microsequencer 302. In the preferred embodiment, microsequencer 302 contains a 3-stage pipeline to perform, for example, fetch, execute, and write operations on the channel program instructions in microcode memory. During the fetch operation, microsequencer 302 retrieves the next channel program

instruction from active microcode memory. During the execute operation, for example, microsequencer 302 executes the previously retrieved channel program instruction. In the preferred embodiment, executing a channel program instruction may include, for example, loading channel program variables from external memory 12 into execution units 500 & 501 (EUs) registers or providing processed output data to external interface processor 14.

[0017] Preferably, EUs 500 & 501 provide cryptographic algorithm support for microsequencer 302. EUs 500 & 501 are configurable elements which perform, among other things, cryptographic combiner type operations on data units. Preferably, EUs 500 & 501 are pre-configured to perform cryptographic combiner type operations to support channel programs executed by associated microsequencer 302. EUs 500 & 501 are preferably pre-configured during a channel program setup and configuration process.

[0018] Among other things, EUs 500 & 501 are each comprised of ALU 502, counter and register elements 503, and CCO 550. Counter and register elements 503 are preferably comprised of hardware including counters, general and dedicated purpose registers, shift registers, and active and shadow data unit address registers. Each ALU 502 and counter and register elements 503 combination perform, among other things, channel program instructions forwarded from microsequencer 302, load and store register operations, arithmetic and bit shift operations including, for example, a logical and of two 32-bit values and left and right bit shift operations. In the preferred embodiment of the present invention, active and shadow data unit address registers are read-only registers by CCE 100. In the preferred embodiment, active and shadow data unit address registers contain an external memory 12 address of a data unit. Preferably, external controller 10 selects one data unit address register as the shadow register for EUs 500 & 501.

[0019] CCO 550 is preferably configured to perform combiner type cryptographic operations. CCO 550 is preferably controlled directly by the coupled ALU 502 and counter and register elements 503 and indirectly by microsequencer 302 and instructions for a channel program performed by ALU 502, counter and register elements 503, and microsequencer 302. In the preferred embodiment, CCO 550 includes, among other things, mode registers 552 & 553, state registers 554 & 555, variable registers 556 & 557, control register 558, status register 560, data in register 564, data out register 566, permuter configuration memories 574 & 575, permuter 576, non-linear function unit (NLFU) configuration memories 579 & 580, and NLFU 578. In cases where there are two of a register type, e.g., mode, state, variable, and data unit address, one of these register sets is titled the active register set and the other is the shadow register set. Also, where there are two of a memory type, e.g., configuration memories for the permuter and con-

figuration memories for the NLFU, one configuration memory for each is titled the active configuration memory and the other is the shadow configuration memory. As described above for active and shadow elements, either register set or configuration memory may be titled active or shadow register set or configuration memory, respectively, as determined by external controller 10. As described below, CCO 550 registers and configuration memories are configured during a setup and configuration process. Although the preferred embodiment is described as having one active and one shadow of each register set and configuration memory, there may be several shadow registers and configuration memories for non-active channel information.

[0020] In the preferred embodiment, the mode register stores, among other things, data used in cryptographic combiner type operations performed by permuter 576 and NLFU 578. Preferably, both active and shadow mode registers are 105-bit registers. Active and shadow state registers 554 & 555 preferably contain channel program state information of the channel program associated therewith. Channel program state is represented by data in the associated 224-bit state register. Active and shadow variable registers 556 & 557 preferably contain channel program variables for the channel program associated therewith. A channel program variable is represented by data in the associated 224-bit variable register. An example of a channel program variable may be a public key encryption key for encrypting plain-text data.

[0021] In the preferred embodiment, control register 558 preferably contains control data which determines, among other things, which elements (e.g., register sets, multiplexers, programmable logic, etc.) are active and which are shadow for CCO 550. In the preferred embodiment, when microsequencer 302 loads control data into control register 558, CCO 550 is signaled to begin pre-configured processing. When processing is complete for pre-configured CCO 550, a status register flag is set which signals microsequencer 302 that CCO 550 is available for further processing. Status register 560 is preferably monitored by microsequencer 302 to determine CCO 550 completion status.

[0022] In the preferred embodiment, microsequencer 302 loads data in register 564. Among other things, data in register 564 stores the next dword, e.g., a 32-bit element of a data unit, which is processed by CCO 550. When CCO 550 processing is complete for a dword, the processed dword is loaded into data out register 566.

[0023] Permuter 576 and NLFU 578 are comprised of hardware and software which are dynamically configurable. When configured, permuter 576 and NLFU 578 may perform cryptographic operations, e.g., symmetric combiner type cryptographic algorithms. In the preferred embodiment, 113 permuter elements are each comprised of a 6-bit control register which configures a 3-stage multiplexer to select 1 of 64 input bits from the associated state register. A 1-bit output from each 113

permuter element is forwarded to NLFU 578. Each of the NLFU's 7 non-linear arrays is comprised of six 12:1, nine 6:1, and seventeen 2:1 NLFUs. Each NLFU array is comprised of configurations of non-linear function units which combine mode, variable, state, and data from a data unit to create a processed dword. In the preferred embodiment, a processed dword may represent, for example, cipher-text, plain-text data, and key stream data.

[0024] In the preferred embodiment, CCE 100 may be fabricated in a multi-chip module, custom cell design, gate array, field programmable gate array, or other technology providing approximately 1.8 million transistors.

[0025] FIG. 2 is a flow chart of a CCE setup and configuration procedure in accordance with a preferred embodiment of the present invention. Procedure 600 is preferably performed by an external controller for, among other things, loading a channel program and hardware configuration information into CCE 100 (FIG. 1). The instructions of the channel program and hardware configuration information are preferably loaded into CCE shadow elements. In the preferred embodiment, procedure 600 is performed in a background staging mode for CCE 100 (FIG. 1).

[0026] In task 602, a channel program is loaded into memory. In the preferred embodiment, an external controller copies a channel program from an external memory into shadow microcode memory. Permuter configuration data and NLFU configuration data are also copied from an external memory into shadow configuration memories and are considered part of the channel program. Preferably, permuter configuration data is copied into a shadow permuter configuration memory and NLFU configuration data is copied into a shadow NLFU configuration memory. An external controller determines which memories are in the shadow state.

[0027] In task 604, configuration data is loaded into a mode register, multiplexer, and programmable logic of CCO 550 (FIG. 1). In the preferred embodiment, an external controller copies the configuration data from an external memory. Configuration data is preferably created by means external to the CCE. Configuration data is characterized by data which configures hardware to perform symmetric combiner type cryptographic operations on data units. Preferably, cryptographic operations are performed under control of the channel program in task 602. Configuration data is preferably comprised of mode, permuter, and NLFU data. Mode data is information used by a channel program in symmetric combiner type cryptographic operations in the NLFU. Mode data is preferably stored in the shadow mode register of CCO 550 (FIG. 1). Permuter data is comprised of data for use in configuring a shadow multiplexer which selects state register output bits. Accordingly, the multiplexer controls data flow from the associated state register into the permuter. NLFU data is comprised of data for use in configuring shadow programmable logic. In the preferred

embodiment, programmable logic is a collection of NLFU arrays which perform operations on 14 dwords, e.g., 7 dwords of state data and 7 dwords of variable data simultaneously.

[0028] In task 606, a channel program context is loaded into a state register, variable register, and data unit address register of EUs 500 & 501 (FIG. 1). The channel program context is comprised of state and variable data associated with a channel program and address data associated with a data unit.

[0029] State data represents a channel program state. In the preferred embodiment, state data may contain, for example, channel program data representing the current state of execution for a process such as encryption. Maintaining the state data allows a channel program to be resumed after being swapped from shadow registers and memories to active registers and memories. Initial state data is preferably written to the shadow state register prior to channel program execution.

[0030] Variable data represents a channel program variable. In the preferred embodiment, variable data may contain, for example, a cryptographic encryption key for use in performing cryptographic operations in CCO 550 (FIG. 1). Initial variable data is preferably written to the shadow variable register prior to channel program execution.

[0031] Address data for a data unit represents the external memory address of a data unit associated therewith. The address references a data unit characterized by, among other things, a header and payload data. Preferably, the header contains two dwords which include, for example, a reference to a destination location. A destination location is an output location where processed dwords are written. Payload data contains, among other things, dwords to be processed by the channel program in task 602. In the preferred embodiment, payload data size is between 1 and 4094 dwords although other sizes are suitable. The address for the data unit is preferably stored in the shadow data unit address register of counter and register elements 503 (FIG. 1).

[0032] FIG. 3 is a flow chart of a data unit processing procedure in accordance with a preferred embodiment of the present invention. Data units are preferably preformatted as discussed above. Data unit processing procedure 700 is preferably performed by CCE 100 (FIG. 1) for, among other things, processing a data unit. Examples of processing a data unit are encryption of plain-text data, decryption of cipher-text data, processing of in-band signaling required for bit, word, or frame formatting, detection of in-band signals, or instruction for operations such as receive over-the-air-rekey (OTAR) data. In the preferred embodiment, the active elements of procedure 700 were titled the shadow elements in procedure 600 (FIG. 2). Also, in the preferred embodiment, procedure 700 is performed concurrently with procedure 600.

[0033] In task 702, the header of the data unit is proc-

essed. In the preferred embodiment, the microsequencer of each control unit fetches and executes instructions of the same channel program from active microcode memory. The microsequencer also performs any write operations external to the CCE as a result of performing instructions. In the preferred embodiment, instructions of the channel program direct the microsequencer to copy the header of the data unit from external memory to a destination location. The destination location is preferably determined by information contained within the header of the data unit. In the preferred embodiment, the destination location is a storage location such as external memory or an external controller or processor. In another embodiment of the present invention, the destination location may be a "temporary" memory location in an external interface processor.

[0034] In task 706, a portion of the payload data is loaded into the data in register. In the preferred embodiment, the payload portion loaded into the data in register is one dword of the payload data. Similar to task 702, the microsequencer fetches, executes, and writes data external to the CCE when performing instructions of the channel program. Preferably, the instructions of the channel program performed in task 706 load a dword into the data in register.

[0035] In task 708, control data is written into the control register. In the preferred embodiment, the microsequencer determines the control data from instructions of the associated channel program. Among other things, control data provides information to the redundant execution units which determines the active register set in each. After the control register is loaded with control data, the microsequencer signals the associated permuter and NLFU to perform their pre-configured cryptographic operations.

[0036] In task 710, a portion of payload data is processed. In the preferred embodiment, the processed payload portion is one dword of the payload data. The dword loaded into data in register 564 (FIG. 1) in task 706 is processed by CCO 550 (FIG. 1). Preferably, the CCO is pre-configured by procedure 600 (FIG. 2) to perform operations on the dword. After the CCO completes its operations, the processed dword is written to data out register 566 (FIG. 1).

[0037] In task 712, a processing complete flag is set in the status register of CCO 550 (FIG. 1). In the preferred embodiment, the microsequencer monitors the associated status register to determine when cryptographic operations are complete for the dword in task 710. Since the microsequencer is both independent and in control of operations performed by the associated CCO, the microsequencer may perform other instructions of the channel program while the associated CCO performs pre-configured operations.

[0038] In task 714, a processed dword is written to output. In the preferred embodiment, when operations on a dword are complete, the processed dword is written to an output destination. Preferably, the output des-

tionation is specified in the header of the associated data unit. In another embodiment, the output destination may be implied by the channel program being executed. In the preferred embodiment, a header may specify, for example, that processed dwords be written to an external memory or interface processor. Preferably, second check logic 151 (FIG. 1) provides an alarm to external controller 10 (FIG. 1) when EUs 500 & 501 (FIG. 1) produce conflicting output values.

[0039] In task 716, a check for additional dwords is performed. In the preferred embodiment, the microsequencer determines the number of dwords in a data unit to process. Preferably, the microsequencer determines the number of dwords to process from payload size information in the header of the data unit. If the microsequencer determines other dwords require processing, task 706 is performed by the microsequencer. If the microsequencer determines no other dwords require processing, task 718 is performed by the microsequencer. In task 718, the microsequencer signals the external controller that processing for a data unit is complete. In the preferred embodiment, the microsequencer signals the external controller by setting a "done" bit in the function register of associated counter and register elements 503 (FIG. 1). The function register is a dedicated purpose register in counter and register elements 503 which, among other things, communicates status information between the redundant microsequencers and external controller.

[0040] In task 720, tasks 702-718 are repeated for other data units. In the preferred embodiment, the CCE performs procedure 700 on additional data units requiring the first channel program described above. Alternatively, a second channel program may be performed on a data unit, without destroying the first channel program instructions, program context, and hardware configuration information. Because the CCE contains shadow and active elements, two separate channel programs, program contexts, and hardware configurations may be maintained by the CCE. Making use of background staging mode, the CCE may perform, among other things, high performance cryptographic processing operations for symmetric combiner type algorithms.

[0041] Thus, an improved cryptographic processing engine and method have been shown. What has also been shown are a configurable, programmable cryptographic processing engine and method suitable for providing interoperability between existing and emerging communications technology. What has also been shown is a cryptographic processing engine suitable for manufacture in commercial semiconductor foundries thereby reducing manufacturing costs. What has further been shown are a cryptographic engine and method which are configurable, programmable, and capable of performing various and changing communications functions. Also shown are a cryptographic engine and method which provide a high security software component with limited accessibility. What has also been

shown are a cryptographic engine and method which rapidly and securely switches programs and context for each data member processed.

[0042] Therefore, the present invention provides:

a configurable cryptographic engine characterized by:
 a first and second microcode memory for storing respective first and second cryptographic programs;
 a microsequencer for processing one of said cryptographic programs; and
 an execution unit having first and second registers for storing respectively, first and second context associated respectively with said first and second cryptographic programs,
 wherein said microsequencer configures said execution unit in response to said first cryptographic program and said execution unit processes a first data unit in accordance with said first context and said first cryptographic program, and
 wherein said second cryptographic program is loaded into said second microcode memory and said second context is loaded into said second registers during the processing of said first data unit for subsequently processing a second data by said execution unit.

[0043] And wherein the microsequencer further configures said execution unit in response to said second cryptographic program and said execution unit processes a second data unit in accordance with said second context.

[0044] And wherein the first and second microcode memories, a control unit and said execution unit are fabricated on a single semiconductor substrate.

[0045] The present invention is further characterized by:

a redundant control unit that includes a redundant microsequencer for processing said one cryptographic program; and
 a redundant execution unit having redundant first and second registers for storing respectively, said first and second context associated respectively with said first and second cryptographic programs, wherein said redundant microsequencer configures said redundant execution unit in response to said first cryptographic program, and said redundant execution unit processes said first data unit in accordance with said first context.

[0046] The present invention is further characterized by:

a first control logic for comparing control unit output data of said control unit with output data of said redundant control unit; and

a second control logic for comparing execution unit output data of said execution unit with execution unit output data of said redundant execution unit.

[0047] Also, the first control logic has means for notifying an external host of discrepancies between said control unit output data of said control unit and said redundant control unit.

[0048] And, the second control logic has means for notifying an external host of discrepancies between said execution unit output data of said execution unit and said redundant execution unit.

[0049] And wherein the execution unit includes a cryptographic co-processor having said first and second registers contained therein, said cryptographic co-processor for performing combiner-type cryptographic operations on said first data unit.

[0050] The present invention also provides, in a configurable cryptographic engine comprised of first and second microcode memories, a microsequencer, an execution unit that includes a cryptographic co-processor, a method for processing data units characterized by the steps of:

loading a first cryptographic program into said first microcode memory;
 loading first configuration bits associated with said first cryptographic program into first registers associated with said execution unit;
 loading first context of said first cryptographic program into second registers associated with said cryptographic co-processor;
 providing, by said microsequencer, first program parameters based on said first cryptographic program to said cryptographic co-processor for configuring said cryptographic co-processor for an operation; and
 performing, by said cryptographic co-processor, as configured with said first program parameters, said operation on a first data unit, said operation determined, at least in part, by said first cryptographic program and said first context.

[0051] And wherein the loading steps are performed during the processing of a second data unit by said cryptographic co-processor in accordance with a second cryptographic program stored in said second microcode memory.

[0052] And wherein second context associated with said second cryptographic program is stored in third registers associated with execution unit, and

second configuration bits associated with said second cryptographic program are stored in fourth registers associated with said execution unit, said second context and said second configuration bits being used by said second cryptographic program for processing said second data unit,

said second data unit being a prior data unit to said first data unit in a series of data units.

[0053] And wherein the first cryptographic program is comprised of a set of instructions, and the method further characterized by the steps of:

said microsequencer retrieving a first instruction from said set of instructions;
 said microsequencer providing said first program parameters based on said first instruction to said cryptographic co-processor to configure said cryptographic co-processor for said operation; and
 performing said operation on a first dword of said first data unit, said operation determined, at least in part, by said first instruction, said first cryptographic program, and said first context.

[0054] The present invention is further characterized by the steps of:

loading a third cryptographic program into said second microcode memory;
 loading third configuration bits associated with said third cryptographic program into said fourth registers;
 loading third context associated with said third cryptographic program into said third registers; and
 processing a third data unit by said execution unit in accordance with said third context and configuration bits after the processing of said first data unit, the steps of loading said third cryptographic program, said third configuration bits, and said third context being performed during the processing of said first data unit.

[0055] The present invention is further characterized by the steps of:

updating said first context in said second registers based on said operation performed on said first data unit;
 repeating the steps of providing, performing said operation and updating said first context for other instructions of said set of instructions; and
 for other words of said first data unit, repeating the steps of retrieving, providing, performing said operation and updating said first context to create an output data unit.

[0056] And, the present invention is further characterized by the step of loading an address of said first data unit into an address register, and reading said first data unit from a storage location indicated by said address.

[0057] And wherein the configurable cryptographic engine is further characterized by a redundant microsequencer and a redundant execution unit, the method further characterized by the steps of:

providing redundant program parameters by said redundant microsequencer to said redundant execution unit in accordance with said second cryptographic program;

processing said first data unit in said redundant execution unit;
 comparing said redundant program parameters with said first program parameters; and
 notifying an external host of discrepancies between said redundant program parameters and said first program parameters.

[0058] The present invention also provides, in a configurable processing engine comprised of first and second microcode memories, a microsequencer, and an execution unit, wherein said first microcode memory has a first program stored therein, a method for processing data units characterized by the steps of:

said microsequencer providing first configuration bits to said execution unit based on said first program;
 said execution unit processing a first data unit based on said first configuration bits and first context associated with said first program, said first context being stored in a first context register associated with said execution unit; and
 loading a second program into said second microcode memory during the processing step.

[0059] The present invention is further characterized by the steps of:

loading second context associated with said second program into a second context register associated with said execution unit during the processing step; and
 said execution unit processing a second data unit based on said second context after the completion of the step of processing said first data unit.

[0060] The present invention is further characterized by the steps of:

loading a third program into said first microcode memory;
 loading third context associated with said third program into said first context register; and
 said execution unit processing a third data unit based on said third configuration bits and third context after the completion of the step of processing said second data unit, wherein
 the loading said third program and third context steps being performed during the step of processing said second data unit.

[0061] Also, the present invention provides a configurable cryptographic processing engine characterized

by:

a first control unit for performing a first channel program wherein said first channel program operates on a first data unit, said first control unit for performing a second channel program wherein said second channel program operates on a second data unit, subsequent to performing said first channel program;
 a first memory for storing said first channel program;
 a second memory for storing said second channel program, said second channel program being loaded in the second memory during the performance of said first channel program; and
 a first execution unit for performing operations on selected portions of said first data unit based on the first channel program, said first execution unit for performing operations on selected portions of said second data unit based on the second channel program, subsequent to performing operations on said first data unit.

The present invention is further characterized by:

a second control unit for performing said first channel program, concurrently during the first control units' performance of the first channel program; and
 a control unit check logic comparing output values of the first and second control units, and providing an alarm signal when the output values conflict.

[0062] And wherein the second control unit for performing said second channel program, concurrently during the first control units' performance of the second channel program.

[0063] And wherein each first and second control unit is comprised of:

a first microsequencer for performing fetch, execute, and write operations of each said first and second channel program, sending control data and receiving status data from said first execution unit.

The present invention is further characterized by:

a second execution unit for performing operations on selected portions of said first data unit, concurrently during the first execution units' performance of operations on selected portions of said first data unit; and
 a execution unit check logic comparing output values of the first and second execution units, and providing an alarm signal when the output values conflict.

[0064] And wherein the second execution unit for performing operations on selected portions of said second

data unit, concurrently during the first execution units' performance of operations on selected portions of said second data unit.

Claims

1. A configurable cryptographic engine (100) characterized by:

a first (200) and second (202) microcode memory for storing respective first and second cryptographic programs;
 a microsequencer (302) for processing one of said first and second cryptographic programs; and
 an execution unit (500) having first and second registers (554 & 555, 556 & 557, 503) for storing respectively, first and second context associated respectively with said first and second cryptographic programs,
 wherein said microsequencer configures said execution unit in response to said first cryptographic program and said execution unit processes a first data unit in accordance with said first context and said first cryptographic program, and
 wherein said second cryptographic program is loaded into said second microcode memory and said second context is loaded into said second registers during processing of said first data unit for subsequently processing a second data unit by said execution unit.

2. A configurable cryptographic engine as claimed in claim 1 wherein said microsequencer further configures said execution unit in response to said second cryptographic program and said execution unit processes a second data unit in accordance with said second context,

and wherein said first and second microcode memories, a control unit and said execution unit are fabricated on a single semiconductor substrate.

3. A configurable cryptographic engine as claimed in claim 1 further characterized by:

a redundant control unit that includes a redundant microsequencer for processing said first cryptographic program;
 a redundant execution unit having redundant first and second registers for storing respectively, said first and second context associated respectively with said first and second cryptographic programs, wherein said redundant microsequencer configures said redundant execution unit in response to said first cryptographic program, and said redundant execution unit processes said first data unit in accord-

ance with said first context;

a first control logic for comparing control unit output data of a control unit with output data of said redundant control unit; and

a second control logic for comparing execution unit output data of said execution unit with execution unit output data of said redundant execution unit,

and wherein said first control logic has a means for notifying an external host of discrepancies between said control unit output data of said control unit and said redundant control unit, and wherein said second control logic has means for notifying an external host of discrepancies between said execution unit output data of said execution unit and said redundant execution unit.

4. In a configurable cryptographic engine (100) comprised of first and second microcode memories (200, 202), a microsequencer (302), an execution unit (500) that includes a cryptographic co-processor (550), a method (600, 700) for processing data units characterized by the steps of:

(602) loading a first cryptographic program into said first microcode memory;

(604) loading first configuration bits associated with said first cryptographic program into first registers associated with said execution unit;

(606) loading first context of said first cryptographic program into second registers associated with said cryptographic co-processor;

(708) providing, by said microsequencer, first program parameters based on said first cryptographic program to said cryptographic co-processor for configuring said cryptographic co-processor for an operation; and

(710) performing, by said cryptographic co-processor, as configured with said first program parameters, said operation on a first data unit, said operation determined, at least in part, by said first cryptographic program and said first context.

5. A method as claimed in claim 4 wherein the loading steps are performed during the processing of a second data unit by said cryptographic co-processor in accordance with a second cryptographic program stored in said second microcode memory,

and wherein a second context associated with said second cryptographic program is stored in third registers associated with said execution unit, and

second configuration bits associated with said second cryptographic program are stored in fourth registers associated with said execution

unit,

said second context and said second configuration bits being used by said second cryptographic program for processing said second data unit,

said second data unit being a prior data unit to said first data unit in a series of data units.

6. A method as claimed in claim 5 wherein said first cryptographic program is comprised of a set of instructions, the method further characterized by the steps of:

said microsequencer retrieving a first instruction from said set of instructions;

said microsequencer providing said first program parameters based on said first instruction to said cryptographic co-processor to configure said cryptographic co-processor for said operation;

performing said operation on a first dword of said first data unit, said operation determined, at least in part, by said first instruction, said first cryptographic program, and said first context;

loading a third cryptographic program into said second microcode memory;

loading third configuration bits associated with said third cryptographic program into said fourth registers;

loading third context associated with said third cryptographic program into said third registers; and

processing a third data unit by said execution unit in accordance with said third context and configuration bits after the processing of said first data unit,

the steps of loading said third cryptographic program, said third configuration bits, and said third context being performed during the processing of said first data unit.

7. In a configurable processing engine (100) comprised of first and second microcode memories (200, 202), a microsequencer (302), and an execution unit (500), wherein said first microcode memory (200) has a first program stored therein, a method (600, 700) for processing data units characterized by the steps of:

(708) said microsequencer providing first configuration bits to said execution unit based on said first program;

(710) said execution unit processing a first data unit based on said first configuration bits and first context associated with said first program, said first context being stored in a first context register associated with said execution unit; and

(602) loading a second program into said second microcode memory during the processing step.

8. A method as claimed in claim 7 further characterized by the steps of: 5

loading second context associated with said second program into a second context register associated with said execution unit during the processing step; 10

said execution unit processing a second data unit based on said second context after the completion of the step of processing said first data unit; 15

loading a third program into said first microcode memory;

loading third context associated with said third program into said first context register; and 20

said execution unit processing a third data unit based on said third configuration bits and third context after the completion of the step of processing said second data unit, wherein the loading said third program and third context steps being performed during the step of processing said second data unit. 25

9. A configurable cryptographic processing engine (100) characterized by: 30

a first control unit (300) for performing a first channel program wherein said first channel program operates on a first data unit, said first control unit for performing a second channel program wherein said second channel program operates on a second data unit, subsequent to performing said first channel program; 35

a first memory (200) for storing said first channel program;

a second memory (202) for storing said second channel program, said second channel program being loaded in the second memory during the performance of said first channel program; and 40

a first execution unit (500) for performing operations on selected portions of said first data unit based on the first channel program, said first execution unit for performing operations on selected portions of said second data unit based on the second channel program, subsequent to performing operations on said first data unit. 45 50

10. A configurable cryptographic processing engine as in claim 9 further characterized by: 55

a second control unit (301) for performing said first channel program, concurrently during the

first control unit's performance of the first channel program,

said second control unit for performing said second channel program, concurrently during the first control unit's performance of the second channel program;

a control unit check logic (150) comparing output values of the first and second control units, and providing an alarm signal when the output values conflict,

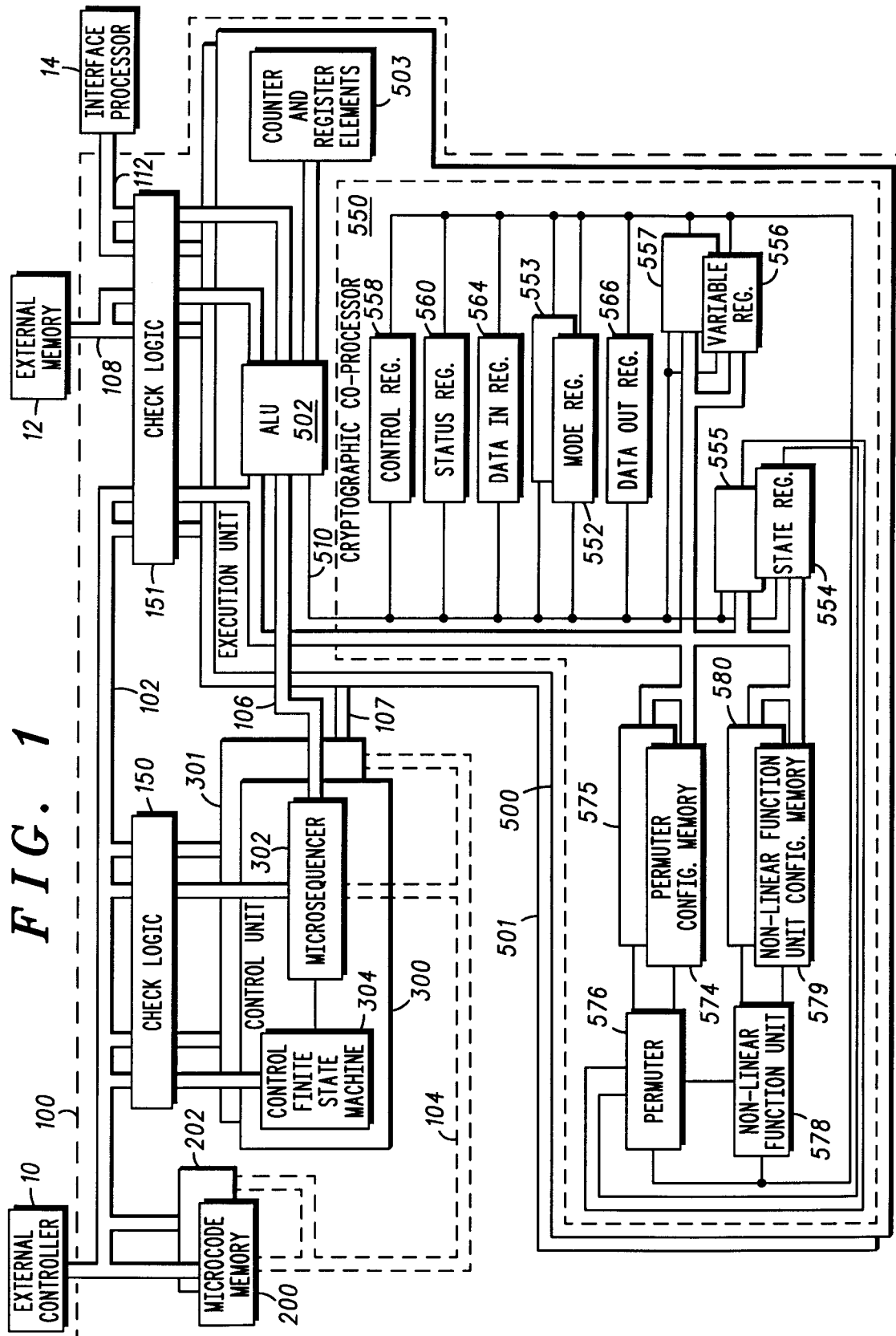
and wherein each first and second control unit is comprised of:

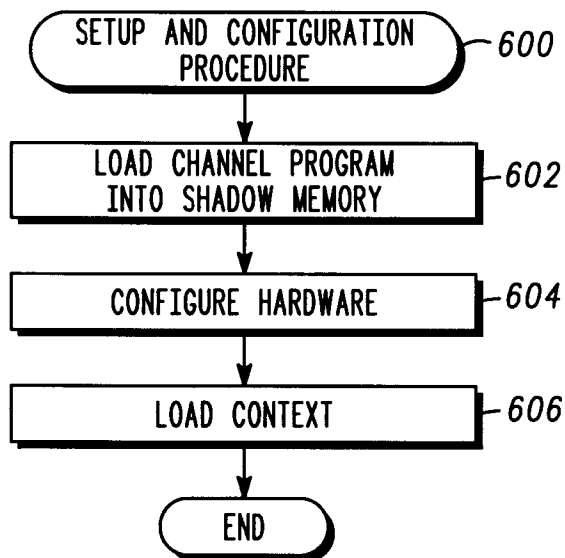
a first microsequencer (302) for performing fetch, execute, and write operations of each said first and second channel program, sending control data and receiving status data from said first execution unit;

a second execution unit (501) for performing operations on selected portions of said first data unit, concurrently during the first execution unit's performance of operations on selected portions of said first data unit; and

a execution unit check logic (151) comparing output values of the first and second execution units, and providing an alarm signal when the output values conflict,

and wherein said second execution unit for performing operations on selected portions of said second data unit, concurrently during the first execution unit's performance of operations on selected portions of said second data unit.



*FIG. 2**FIG. 3*